

Validation de la sémantique d'un modèle semi-formel de connaissances avec OntoCASE

Michel Héon¹, Josianne Basque¹, Gilbert Paquette¹

¹ Centre de recherche LICEF, Télé-Université, Montréal, Canada
(michel.heon, josianne.basque, gilbert.paquette)@licef.ca

Résumé : L'activité de modélisation de connaissances consiste à élaborer une représentation externe structurée de la conceptualisation subjective qu'a un agent cognitif (le modélisateur) d'un domaine de connaissances. Cette conceptualisation est externalisée dans un « modèle » à l'aide d'un langage qui peut être plus ou moins formel. Pour un agent cognitif humain, l'usage d'un langage semi-formel favorise l'expressivité au cours de l'activité de modélisation, tout en réduisant l'ambiguïté du langage informel. Un modèle, qu'il soit informel, semi-formel ou formel peut néanmoins comporter des erreurs de représentation. La production automatique par un agent informatique de la liste des propositions sémantiques exprimées en langage naturel et résultant d'une interprétation formelle d'un modèle semi-formel pourrait guider le modélisateur à objectiver sa conceptualisation du domaine. Cet article présente comment OntoCASE, une méthodologie d'ingénierie ontologique, peut aider le modélisateur pour ce faire. Nous décrivons les principaux composants de cette méthodologie et exposons quelques exemples de modèles semi-formels comportant des ambiguïtés ou des erreurs de représentation afin d'illustrer comment OntoCASE assiste le modélisateur à objectiver sa conceptualisation. Nous présentons aussi quelques résultats d'une expérimentation d'OntoCASE par quelques modélisateurs permettant de démontrer son potentiel en tant qu'outil de validation de modèles semi-formels.

Mots-clés : Modélisation, ingénierie ontologique, interprétation automatique, modèle semi-formel, validation de modèles.

1 Introduction

Pour catégoriser le degré de formalisation d'un langage, Uschold (1996) propose une classification en quatre degrés: informel, semi-informel, semi-formel et formel. Un langage semi-formel peut être défini comme un langage artificiel dont la sémantique comporte des éléments d'ambiguïté. Par exemple, un langage de pseudocode informatique est une sorte de langage semi-formel.

Pour modéliser, le modélisateur doit *analyser et abstraire* la partie de la réalité, qu'il désire modéliser afin d'évoquer dans son esprit une *conceptualisation*. Le contenu de cette conceptualisation est schématisé dans un modèle à l'aide d'un langage. En modélisation semi-formelle, le modélisateur utilise un langage qui comporte dans sa sémantique la propriété d'être *polysémique*. La polysémie fait référence à la propriété d'un symbole de représenter plus d'un objet. Par exemple, le phonème [mo] peut représenter le terme "maux" ou le terme "mot". Comparativement

aux langages formels, qui par définition ne possèdent pas de polysémie, les langages semi-formels offrent une plus grande facilité d'utilisation grâce à cette propriété.

Parfois, un modèle contient des *erreurs de représentation*. On peut définir une erreur de représentation comme une inadéquation entre le modèle produit et la réalité qu'il tente de représenter. Ces erreurs de représentation peuvent survenir pendant la phase d'abstraction de la réalité à conceptualiser et/ou pendant la phase d'expression de la conceptualisation dans le modèle. Dans le premier cas, les erreurs surviennent par une compréhension erronée de la réalité conceptualisée ou par une conceptualisation inadéquate par rapport à l'objectif de modélisation, alors que dans le deuxième cas, les erreurs peuvent provenir d'une mauvaise utilisation du langage de modélisation (vocabulaire et grammaire) ou d'une compréhension fautive de sa sémantique. Par exemple, un modélisateur qui, après avoir analysé un processus obtient la conceptualisation suivante: "*L'élimination des déchets se fait de deux façons principales: l'incinération et l'enfouissement*". Il peut alors exprimer dans un modèle cette conceptualisation de deux façons différentes. Soit il utilise un méronyme (part-of) pour représenter que *incinération* et *enfouissement* font "partie de" *éliminer des déchets*; soit il utilise un hyponyme (is-a) pour représenter qu'*incinérer* et *enfouir* sont des "sorte de" *élimination de déchets*. Sans analyse objective de ces modèles, il est difficile de déterminer lequel de ces deux modèles représente le mieux le processus. Pour répondre à cette question, nous proposons d'utiliser OntoCASE et sa capacité à produire un texte en langage naturel qui représente une interprétation formelle du modèle semi-formel. À la lecture du texte produit, le modélisateur sera en mesure de comparer ce texte avec la conceptualisation qu'il se fait de la réalité qu'il tente de représenter. S'il y a discordance, le modélisateur peut s'interroger si elle provient d'une mauvaise conceptualisation de la réalité ou d'une mauvaise utilisation du langage pendant l'expression de cette conceptualisation.

La modélisation d'un domaine de connaissances exprimée à l'aide d'un langage semi-formel offre de multiples avantages. La plus grande souplesse d'expression d'un tel langage par rapport aux langages hautement formels utilisés par exemple pour construire des ontologies, surtout s'il s'appuie sur un formalisme graphique, favorise le processus d'explicitation de connaissances tacites (Basque et Pudelko, 2010a). Dans un tel cadre de modélisation semi-formelle, la spontanéité n'est pas bloquée par la charge cognitive associée à une formalisation trop poussée de la pensée (Basque *et al.*, 2010a); la charge cognitive est alors davantage dévolue à l'activité de schématisation plutôt qu'à celle de formalisation. De plus, par rapport à un langage strictement formel, l'usage d'un langage semi-formel offre la possibilité : a) d'élargir le bassin des personnes aptes à représenter leurs connaissances, sans l'aide d'un ingénieur de connaissances; b) dans un contexte de gestion de connaissances dans les organisations, d'économiser du temps pour l'élicitation des connaissances expertes, libérant ainsi plus rapidement les experts pour la réalisation de leur travail habituel; c) de réaliser un gain de qualité dans la représentation des connaissances, dans la mesure où le langage semi-formel possède une plus grande expressivité, permettant par exemple de représenter des connaissances procédurales, ce que ne permettent pas la plupart des langages formels.

Le langage semi-formel comporte aussi des avantages par rapport à un langage informel car il fournit un certain *guidage représentationnel* (Suthers, 2003) qui structure le processus de modélisation. Le modèle qui en résulte constitue une première itération de formalisation.

OntoCASE est une méthodologie de conception d'ontologies dans le formalisme *Ontology Web Language* (OWL) fondée sur une modélisation semi-formelle préalable. Cette méthodologie est assistée par un système expert à la formalisation qui, pendant la phase de validation de l'ontologie cible, produit un rapport de validation sémantique en langage naturel. Afin de produire ce rapport, le système expert d'OntoCASE applique des inférences à l'ontologie cible et en traduit les conclusions en langage naturel. Nous proposons d'utiliser le contenu de ce rapport résultant d'une interprétation automatisée et objective de la sémantique du modèle d'origine pour amener le modélisateur à objectiver sa conceptualisation du domaine telle qu'exprimée dans le modèle semi-formel.

Le but de cet article est donc de présenter comment l'utilisation d'OntoCASE permet de supporter une telle démarche. De manière plus spécifique, nous verrons comment OntoCASE permet de désambiguïser certaines représentations semi-formelles, telles celle d'une relation de *composition* versus une relation de *spécialisation* entre des connaissances; celle de la connaissance *factuelle* versus la connaissance *abstraite* et celle de la relation de *précédence* entre connaissances procédurales et stratégiques. De plus, nous rapportons une expérimentation en laboratoire qui a permis de démontrer l'efficacité d'OntoCASE en tant qu'assistant à la modélisation et de suggérer son potentiel en tant qu'outil de validation sémantique de modèles exprimés en langage semi-formel.

2 La polysémie du langage MOT

Le langage MOT (Paquette, 2002b) s'est avéré efficace en tant que système de représentation graphique semi-formelle de connaissances autant dans un contexte d'ingénierie pédagogique (Paquette, 2002a, 2010) que dans celui de la gestion de connaissances dans des organisations (Basque *et al.*, 2010a) de même que dans un contexte éducatif (Basque et Pudelko, 2010b).

La polysémie du langage MOT, intrinsèque à ce type de langage, est une caractéristique qu'exploitent ses utilisateurs. De fait, sa polysémie en constitue l'un des vecteurs de son acceptation et de son utilisation par les usagers. Cependant, la polysémie peut parfois donner lieu à une certaine ambiguïté sémantique quand vient le temps d'interpréter un modèle construit avec ce langage. La table 1 résume les différentes significations pouvant être associées aux entités et aux relations constituant les primitives du langage. MOT différencie les types de connaissances au moyen de symboles graphiques catégorisés en *entités* et en *relations*. Les connaissances peuvent être combinées au sein d'un même diagramme de manière à produire des modèles représentant les relations entre connaissances de divers types. À l'instar des théories sur la représentation des connaissances, le langage MOT offre la possibilité de représenter des connaissances selon deux niveaux d'abstraction, soit d'une part le niveau abstrait (*ConnaissanceAbstraite*) qui inclut les connaissances

conceptuelles, procédurales et stratégiques, et d'autre part le niveau factuel (*Fait*), ce qui en langage des descriptions correspond à la TBOX et la ABOX (Baader, Calvanese *et al.*, 2007).

Table 1. Polysémie des entités et des relations du langage MOT

a) Polysémie des entités		b) Polysémie des relations	
Représentation en MOT	Polysémie	Représentation en MOT	Polysémie
Niveau abstrait	<div style="border: 1px solid black; padding: 2px; display: inline-block;">concept</div> Connaissance conceptuelle	Classe	
		Schéma	Entier
			Caractère
			Naturel
	<div style="border: 1px solid black; border-radius: 50%; padding: 2px; display: inline-block;">procédure</div> Connaissance procédurale	Action	
		Opération	
	<div style="border: 1px solid black; padding: 2px; display: inline-block;">principe</div> Connaissance stratégique	Agent	
		Propriété	
		Règle	
		Condition	
Niveau factuel	<div style="border: 1px dashed black; padding: 2px; display: inline-block;">exemple</div> Connaissance déclarative	Objet	
		Valeur	
	<div style="border: 1px dashed black; border-radius: 50%; padding: 2px; display: inline-block;">trace</div> Connaissance procédurale	Acte	
		Instruction	
	<div style="border: 1px dashed black; padding: 2px; display: inline-block;">énoncé</div> Connaissance stratégique	Agent	
		Assertion	
		Condition	
		Règle	
	--C--> Composition	Est composé de	
		A pour attribution	
-IP-> Intrant/Produit		A pour intrant	
		A pour produit	
--R--> Régulation		Régulation	
--P--> Précédence		Puis exécuter	
	Puis évaluer		
	Permet		
--I--> Instance	Évaluer à partir de		
	A pour dépendance		
	A pour instance		
--S--> Spécialisation	Est une sorte de		
--A--> Application	A pour application		

Du point de vue de la sémantique, la connaissance de type *concept* représente « le quoi » des choses. Elle sert à décrire l'essence d'un objet concret. Le concept peut être associé à l'idée de classe. En ce sens, il est l'abstraction d'un objet concret. Dans MOT, le concept est aussi utilisé pour représenter le type (le schéma) d'un attribut associé à un concept. L'attribut peut ainsi être désigné comme étant un nombre entier, réel ou une chaîne de caractères. *L'exemple* représente l'objet d'un concept en énonçant un certain nombre de faits qui le décrivent. L'exemple est aussi utilisé pour représenter une *valeur* associée à un schéma. La *procédure* permet de décrire « le comment » des choses. Elle désigne tout autant des opérations et des actions pouvant être accomplies par l'humain que des phénomènes dynamiques. La *trace* représente

l'ensemble des faits concrets obtenus lors de l'exécution d'une procédure, soit un *acte* ou une *instruction*. Le *principe* désigne « le pourquoi », « le quand » ou le « qui » associé à une chose. Il est une connaissance stratégique (Paquette, 2002b ; Paris, Lipson *et al.*, 1983) qui permet d'énoncer des propriétés qui existent entre des objets, que ce soit des concepts, des procédures ou d'autres principes. Il sert aussi à représenter une condition pouvant s'appliquer à l'exécution d'une action ou à représenter un agent qui agit sur quelque chose. Le principe est aussi utilisé pour construire une règle. L'*énoncé* représente l'instanciation d'un principe à propos d'objets concrets.

La *relation* est un lien unidirectionnel qui unit des connaissances (voir la table 1b). Le langage MOT offre un ensemble de liens qui sont typés et qui sont dans certains cas polysémiques. La table 2 présente les règles d'utilisation des divers liens du langage. Le lien d'*instanciation* (LienI) associe à une connaissance abstraite l'ensemble des faits qui caractérisent une instance de cette connaissance. Le lien d'instanciation n'est pas une relation transitive. Le lien *intranant/produit* (LienIP) sert à associer une connaissance procédurale à une connaissance conceptuelle afin de représenter l'intrant ou le produit d'une procédure. Ce lien est notamment utile dans la description des algorithmes, des processus et des méthodes. Le lien intrant/produit n'est pas une relation transitive. Le lien de *précédence* (LienP) associe une connaissance à une autre qui la suit dans une séquence temporelle de procédures ou de règle de décision (principes). Dans son caractère transitif, le lien de précédence détermine les procédures qui sont permises et celles de qui elles dépendent. Dans son caractère non transitif, le lien de précédence change de signification selon qu'il indique la précédence d'une procédure (puis-exécuter) ou une condition (puis-évaluer). Le lien de *régulation* (lienR) associe une connaissance stratégique (un Principe ou un Énoncé) à une autre connaissance afin de préciser une contrainte, une restriction ou une règle qui régit la connaissance. Le lien de *composition* (LienC) permet de représenter l'association entre une connaissance et des connaissances qui la composent. Le lien de composition est une relation transitive. Le lienC est aussi utilisé pour associer un attribut à un concept.

Table 2. Règle d'utilisation des relations en langage MOT

Destination Origine	Connaissances abstraites			Connaissances factuelles		
	Concept	Procédure	Principe	Exemple	Trace	Énoncé
Concept	C, S	I/P	R	I, C		
Procédure	I/P	C, S, P	C, P		I, C	
Principe	R	C, R, P	C, S, P, R			I, C
Exemple	A	A	A	A, C	A, I/P	A
Trace	A	A	A	A, I/P	A, C, P	A, C, P
Énoncé	A	A	A	A, R	A, C, R, P	A, C, R, P

3 L'outil OntoCASE

Originellement conçu pour la construction d'ontologies à partir d'un modèle semi-formel, OntoCASE (Héon et Paquette, 2010 ; Héon, Paquette *et al.*, 2009a) est ici utilisé pour générer, en langage naturel, la liste des propositions inférées à partir d'un modèle semi-formel à la suite d'une transformation de nature ontologique.

L'architecture méthodologique d'OntoCASE est représentée à la figure 1. À l'instar de Gómez-Pérez *et al* (2003), nous définissons une méthodologie comme étant un ensemble de méthodes dont chacune se réalise en appliquant un ensemble de techniques. Dans notre cas, elle est supportée par un agent informatique intelligent. Trois méthodes assistées composent la méthodologie d'OntoCASE:

1) La méthode *Concevoir un modèle semi-formel du domaine* est supportée par l'éditeur de modèles eLi (Héon *et al.*, 2009a). La méthode produit un modèle semi-formel du domaine, prêt à être formalisé. Pour guider sa modélisation, le modélisateur utilise, le cas échéant, les rapports de validation syntaxique et sémantique obtenus lors d'une itération précédente. De même, le modélisateur ajustera sa modélisation en fonction des messages d'erreurs qu'il aura obtenus lors d'une formalisation antérieure.

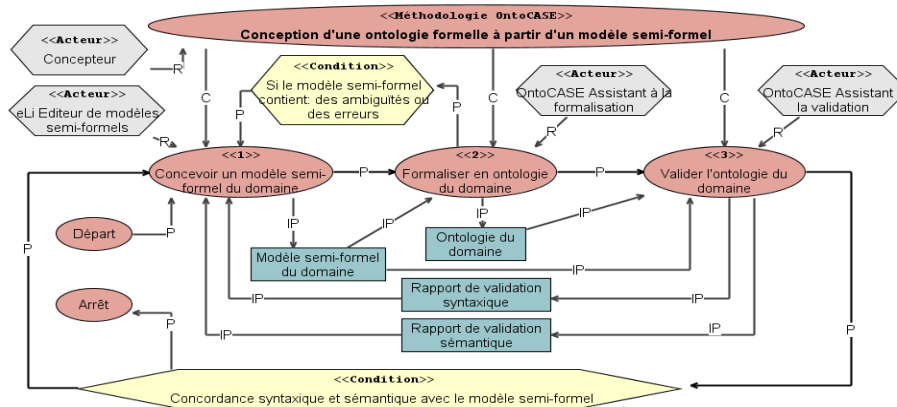


Fig. 1 –Méthodologie OntoCASE.

2) La méthode *Formaliser en ontologie du domaine* est supportée par un assistant à la formalisation qui génère l'ontologie du domaine. Trois processus composent cette méthode, soit les processus *importer*, *désambiguïser* et *convertir* (Héon *et al.*, 2009a). Le processus d'importation traduit le modèle semi-formel exprimé dans la notation native à l'éditeur de modèles semi-formels (ici eLi) en modèle dans la notation de l'*Ontology Web Language* (OWL). Le processus de désambiguïsation a pour objectif de lever la polysémie du langage afin de produire un modèle dont les éléments sont non ambigus. Un ensemble de dispositifs (menus déroulants, pointage dans le schéma du modèle, messages textuels publiés à la console) sont autant de mécanismes d'interaction qui assurent la fluidité de la désambiguïsation et qui incitent le modélisateur à s'interroger sur des éléments du langage qu'il utilise. Lorsque la désambiguïsation est terminée, le modélisateur convertit le modèle désambiguïsé en ontologie OWL grâce au module de conversion qui est intégré à OntoCASE.

3) La méthode *Valider l'ontologie du domaine* est assistée par le module de validation, qui produit, entre autres, une interprétation en langage naturel du modèle d'origine. La conception de l'ontologie est un processus itératif qui prend fin lorsque la sémantique du modèle d'origine correspond à la sémantique de l'ontologie cible. Pour aider le modélisateur à procéder à la validation de la sémantique du modèle, le module de validation produit un rapport de validation sémantique prenant la forme de la liste des propositions du modèle semi-formel exprimées en langage naturel et qui

ont été déduites de ce modèle par un moteur d'inférence. Ce rapport sert d'intrant à une nouvelle itération de la conception du modèle semi-formel.

4 Démonstration et expérimentation d'OntoCASE

Nous présentons d'abord une démonstration de la manière dont OntoCASE permet de désambigüiser une représentation prenant la forme d'une combinaison de concepts. Cette démonstration sera suivie par celle d'une désambigüisation de connaissances de nature procédurale. Finalement, nous présentons le protocole utilisé en laboratoire pour démontrer l'efficacité d'OntoCASE en tant qu'outil d'ingénierie ontologique de formalisation d'un modèle semi-formel, qui a permis de révéler son potentiel en tant qu'outil de validation de ce dernier.

4.1 La combinaison de concepts

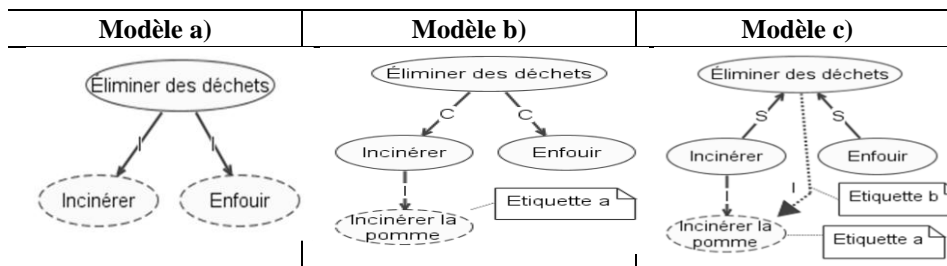
Soit la conceptualisation suivante: *L'élimination des déchets se fait de deux façons principales: l'incinération et l'enfouissement*. La table 3 présente trois façons adéquates et différentes de schématiser cet énoncé dans un modèle semi-formel. Les interprétations générées par OntoCASE apparaissent sous les modèles. Le choix de l'un ou de l'autre de ces modèles s'effectue en visant l'adéquation entre le modèle et la réalité conceptualisée. Cette adéquation est réalisée en comparant l'interprétation du modèle produit par OntoCASE avec l'interprétation humaine de la réalité modélisée.

Le modèle **a** est une première représentation possible de l'énoncé. Le paradigme mis en évidence dans cette représentation est le changement de niveau d'abstraction qui est représenté par l'utilisation du LienI. Dans ce modèle, *Éliminer des déchets* est une connaissance procédurale relevant du niveau abstrait alors qu'*incinérer* et *enfouir* sont définies ici en tant que connaissances du niveau factuel, c'est-à-dire en tant qu'actions concrètes qui sont catégorisées par la connaissance abstraite *Éliminer des déchets*. Les interprétations générées par OntoCASE aux lignes **1** et **2** indiquent bien cet état de fait par l'utilisation de la relation *est-un* entre la *trace* désignée entre crochet « [] » et le *processus* désigné entre parenthèses « () ». Dans ce cas de figure, la lecture de ces interprétations indique au modélisateur qu'*Incinerer* et *Enfouir* sont des constituants d'une énumération dont *Éliminer des déchets* est l'abstraction.

Une autre représentation possible repose sur l'utilisation du méronyme (LienC) à un niveau abstrait. Le modèle **b** est un exemple de schématisation de cette conceptualisation. L'interprétation formelle de ce modèle indique que la procédure *Éliminer de déchets* se compose des procédures *Incinerer* et *Enfouir*. Pour vérifier le comportement du méronyme, nous pouvons utiliser la technique de test par la *vérification d'instance* (Mariot, Golbreich *et al.*, 2008) qui consiste à ajouter une instance fictive au modèle afin de vérifier les conclusions produites après avoir réalisé une inférence sur le modèle. La connaissance factuelle *incinérer une pomme* (voir l'étiquette a) a ainsi été introduite au modèle. À la lecture de son interprétation automatique par OntoCASE, on constate que le fait est une instance d'*Incinerer* mais pas d'*Éliminer les déchets*. Ce qui dans certains contextes pourrait être considéré comme une erreur de représentation qui doit être envisagé par le modélisateur.

Le modèle en **c** utilise l'hyponyme (LienS) pour lier la procédure principale aux sous-procédures pour ainsi former un modèle taxonomique. Là aussi, nous avons appliqué la *vérification d'instance*. Il est constaté à la ligne 9 et à l'étiquette **b** qu'OntoCASE conclut qu'*Incinérer une pomme* est une connaissance factuelle appartenant à *Incinérer* et à *Éliminer les déchets* (ce qui n'est pas le cas dans le modèle **b**). Le modèle **c** constitue ainsi une représentation sémantiquement plus étendue.

Table 3. Divers modèles MOT de l'énoncé : *L'élimination des déchets se fait de deux façons principales: l'incinération et l'enfouissement.*



Interprétation automatique du modèle a)

1. [Enfouir] est un (Éliminer des déchets)
2. [Incinérer] est un (Éliminer des déchets)

Interprétation automatique du modèle b)

3. Éliminer des déchets se compose de Enfouir
4. Éliminer des déchets se compose de Incinérer
5. [Incinérer la pomme] est un (Incinérer)

Interprétation automatique du modèle c)

6. Enfouir est une sorte de Éliminer des déchets
7. Incinérer est une sorte de Éliminer des déchets
8. [Incinérer la pomme] est un (Incinérer)
9. [Incinérer la pomme] est un (Éliminer des déchets)

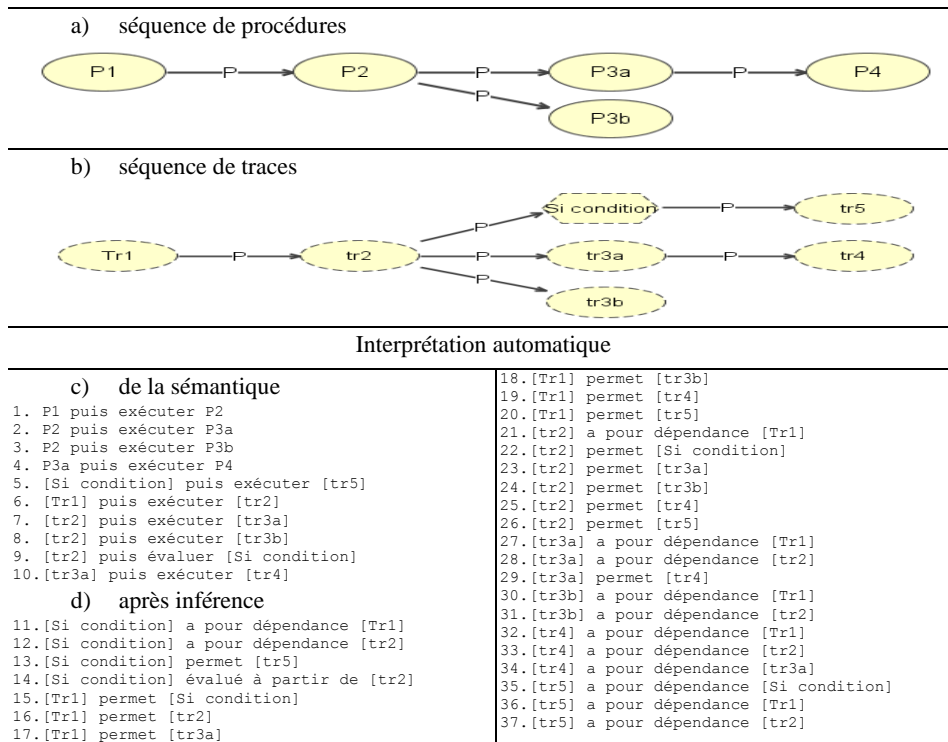
Cette démonstration met en évidence trois manières de représenter l'idée de combinaison de concepts, soit : (1) par le changement de niveau d'abstraction, (2) par l'utilisation du méronyme et (3) par l'utilisation de l'hyponyme. L'interprétation automatique et la *vérification d'instance* du modèle sont des moyens de diagnostic du modèle mis à la disposition du modélisateur. Ces moyens permettent au modélisateur d'ajuster convenablement la représentation exprimée dans le modèle afin de comparer l'interprétation automatique du modèle à l'interprétation attendue de la conceptualisation qu'il se fait de la réalité.

4.2 Modélisation de connaissances procédurales

L'utilisation de connaissances procédurales et de liens de précedence permet de représenter une séquence d'exécution. La table 4 présente en **a** un modèle d'une séquence de procédures qui représente le déroulement d'une séquence de catégories d'actions. Le modèle présenté en **b** représente la même séquence mais au niveau factuel : il s'agit ici d'actions concrètes qui, dans le vocabulaire de MOT, sont appelées des *traces*. Ce modèle intègre aussi un *énoncé* (le polygone) qui, dans ce contexte, symbolise une condition à réaliser pour exécuter le reste de la séquence. Les interprétations combinées de ces deux modèles, telles que générées par OntoCASE,

sont présentées en **c** (avant inférence) et en **d** (après désambiguïsation et inférence). L'interprétation après inférence (en **d**) constitue une sorte de bilan des étapes précédentes et subséquentes à chacun des éléments procéduraux du modèle. Par exemple, l'exécution de *tr3a* a pour dépendance l'exécution de *tr2* et *tr1* (voir les lignes 27 et 28) et permet l'exécution de *tr4* (voir la ligne 29). Ainsi, cette séquence se lit : *tr1 puis exécuter tr2 puis exécuter tr3a puis exécuter tr4* (voir les lignes 6, 7 et 10).

Table 4. Représentation d'une séquence d'actions procédurales et de traces.



4.3 Expérimentation d'OntoCASE

Afin d'évaluer l'utilisabilité d'OntoCASE, quatre participants familiers avec le langage MOT ont été sollicités pour participer, de manière individuelle, à une expérimentation. Deux d'entre eux étaient non familiers avec la conception d'ontologies. La séance d'expérimentation a été divisée en quatre phases: la première phase, la *familiarisation*, avait pour objectif d'initier le participant aux aspects méthodologiques et informatiques d'OntoCASE. La deuxième phase, la *formalisation*, avait pour objectif d'initier le participant au processus de formalisation et de validation de modèles et de permettre à celui-ci de réaliser une première manipulation. Un modèle semi-formel déjà conçu, identique à celui présenté dans Héon *et al* (2009b), était alors présenté au participant qui devait le formaliser en

Le texte présenté aux participants en phase 3 nécessite l'utilisation de connaissances tant déclaratives, procédurales que stratégiques. Le modèle présenté à la table 5 est l'une des représentations possibles de ce texte. Le modèle figurant en **a** représente le processus de haut niveau de la procédure de gestion des déchets, alors qu'en **b**, on retrouve le sous-modèle de la procédure *Incinérer*. L'interprétation automatique de la sémantique du modèle (présentée en **c**) sous la forme de propositions textuelles peut être comparée à celles contenues dans le texte d'origine. La production de la version finale de ce modèle est obtenue après quelques itérations du processus de *Modélisation semi-formelle* -> *Transformation du modèle en ontologie* -> *Génération automatique du texte* -> *Adaptation du modèle*.

L'observation des participants au cours de leur utilisation d'OntoCASE de même que l'analyse de leurs commentaires ont permis de montrer que, dans son état actuel, OntoCASE est un outil convivial qui supporte le concepteur dans l'activité de modélisation et de validation de la sémantique du modèle semi-formel. La rétroaction fournie par l'assistant informatique est utile pour la désambiguïsation des primitives du langage semi-formel ainsi que pour l'objectivation de la conceptualisation du concepteur.

5 Discussion

Bien qu'originellement conçu pour formaliser des modèles semi-formels en ontologies, OntoCASE peut aussi être utilisé pour soutenir un processus de validation sémantique d'un modèle semi-formel. L'étape de validation d'OntoCASE produit un rapport de validation sémantique qui est, en fait, une interprétation formelle du modèle semi-formel d'origine. L'interprétation formelle du modèle tient compte de la nature déclarative, procédurale et stratégique des connaissances que le modèle peut représenter. L'interprétation formelle du modèle semi-formel d'origine intrinsèque à la méthodologie OntoCASE, offre au modélisateur un mécanisme d'objectivation du modèle semi-formel. Par la rétroaction que procure OntoCASE et par l'application de la *vérification d'instances*, le modélisateur peut évaluer la qualité de la sémantique du modèle et y apporter des changements qui lui apparaissent nécessaires.

L'expérimentation en laboratoire a aussi démontré que l'usage d'OntoCASE peut être utile à des non-spécialistes de la modélisation ontologique. Bien que l'objectif visé au cours de cette expérimentation fût la conception d'une ontologie, nous avons pu observer que la rétroaction fournie par OntoCASE était utile pour permettre à ces derniers d'objectiver leur conceptualisation du domaine de connaissances ciblé. Certains participants ont indiqué que cette expérience a modifié certains de leurs paradigmes de modélisation semi-formelle avec MOT.

Nous pensons qu'OntoCASE pourrait servir dans divers contextes d'application, notamment dans des contextes éducatifs afin de permettre aux apprenants d'objectiver leurs conceptualisations de divers domaines de connaissances. En élicitation des connaissances, OntoCASE pourrait permettre de valider la représentation semi-formelle de l'expertise d'un expert ou encore de générer une représentation textuelle en langage naturel de cette expertise. En transfert d'expertise, l'utilisation de l'outil d'inférence pour générer de nouvelles connaissances de domaines pourrait servir de

tuteur pour le développement de nouvelles connaissances chez les novices. Des recherches devront toutefois démontrer l'efficacité d'OntoCASE dans ces divers contextes.

6 Référence

- BAADER F., CALVANESE, D., MCGUINNESS, D. L., NARDI, D. & PATEL-SCHNEIDER, P. F. (2007). *The Description Logic Handbook*: Cambridge University Press.
- BASQUE J. & PUDELKO, B. (2004). *First International Conference on Concept Mapping* (Pamplona, September 14-17). A. J. Canas, J. D. Novak et F. M. Gonzalez Universidad Publica de Navarra, 67-74 p.
- BASQUE J. & PUDELKO, B. (2010a). «La comodélisation des connaissances par objets typés: Une stratégie pour favoriser le transfert d'expertise dans les organisations.». *Revue Télescope*, no spécial: Le transfert intergénérationnel des connaissances.
- BASQUE J. & PUDELKO, B. (2010b). Modeling for Learning. Dans, Gilbert Paquette Eds. *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies*. Hershey, New York: IGI Global.
- GÓMEZ-PÉREZ A., FERNÁNDEZ-LÓPEZ, M. & CORCHO, O. (2003). *Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, First edition. New York: Springer.
- HÉON M. & PAQUETTE, G. (2010). From semi-formal Models to Formal Models. Dans, Gilbert Paquette Eds. *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies*. Hershey, New York: IGI Global.
- HÉON M., PAQUETTE, G. & BASQUE, J. (2009a). «Conception assistée, d'une ontologie à partir d'une conceptualisation consensuelle exprimée de manière semi-formelle». Dans *Conférence d'Ingénierie des connaissances* (25 au 29 mai 2009).
- HÉON M., PAQUETTE, G. & BASQUE, J. (2009b). «Illustration d'un scénario d'assistance à la conception d'une ontologie à partir d'un modèle semi-formel». Dans *Conférence d'Ingénierie des connaissances* (25 au 29 mai 2009).
- MARIOT P., GOLBREICH, C., COTTON, J., VEXLES, F. & BERGER, A. (2008). Méthode, Modèle et outils Ardams de capitalisation des connaissances. Dans, Cépaduès Eds. *Modélisation des connaissances*. Toulouse, France.
- PAQUETTE G. (2002a). *L'ingénierie pédagogique*. Sainte-Foy (Québec): Presses de l'Université du Québec.
- PAQUETTE G. (2002b). *Modélisation des connaissances et des compétences : un langage graphique pour concevoir et apprendre*. Sainte-Foy: Presses de l'UQ.
- PAQUETTE G. (2010). *Visual Knowledge and Competency Modeling - From Informal Learning Models to Semantic Web Ontologies*. Hershey, PA: IGI Global.
- PARIS S., LIPSON, M. Y. & WIXSON, K. K. (1983). «Becoming a Strategic Reader». *Contemporary Educational Psychology*. vol. 8, p. 293-316.
- SUTHERS D. D. (2003). Representational guidance for collaborative inquiry. Dans, J. Andriessen, M. Baker et D. Suthers Eds. *Arguing to Learn* p. 27-46. Dordrecht/Boston/London: Kluwer.
- USCHOLD M. & GRUNINGER, M. (1996). «Ontologies: Principles, Methods and Applications». *Knowledge Engineering Review*. vol. 11, no 2, p. 93-136.